DUAL: CLOSING THE PLAN–EXECUTION GAP BETWEEN MLLMS AND DITS FOR VIDEO EDITING

Anonymous authors
Paper under double-blind review

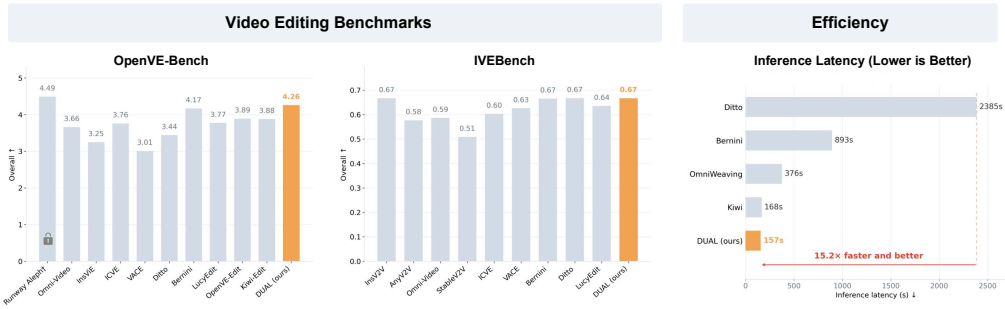


Figure 1: DUAL outperforms prior methods on video editing benchmarks while remaining efficient. Left: quality on OpenVE-Bench and IVEBench. Right: latency and renderer parameter count.

ABSTRACT

Strong multimodal understanding and high-quality video generation do not guarantee faithful video editing: a model may identify the requested change yet edit the wrong object, assign the wrong attribute, or violate the relation that identifies the target. We call this mismatch the *plan–execution gap* and trace it to the interface between a multimodal large language model (MLLM) and a diffusion transformer (DiT). We propose **DUAL**, which routes target-state information and relational constraints through different paths. Its Target-State Path converts MLLM features from multiple layers into tokens that join source and noisy target tokens in DiT self-attention, while its Relational-Constraint Path combines final-layer MLLM features with native text embeddings for cross-attention guidance. DUAL reaches the highest Overall score among the compared open methods on both IVEBench (0.67) and OpenVEBench (4.26), while running up to 15.2 times faster than the slowest compared baseline in average inference latency. These results support separating target-state information, which should participate in denoising, from relational constraints, which should guide it.

1 INTRODUCTION

Multimodal large language models (MLLMs) and diffusion transformers (DiTs) have complementary strengths for video editing (Bai et al., 2025b;a; Peebles & Xie, 2023; Wan et al., 2025). MLLMs can reason about objects, attributes, relations, and the intended outcome of an instruction, while DiTs can synthesize realistic and temporally coherent video (Bai et al., 2025b; Lin et al., 2026a; Wan et al., 2025; Kong et al., 2025; Wu et al., 2025a; HaCohen et al., 2026). A natural system therefore lets an MLLM plan an edit and lets a DiT render it (Team et al., 2026; Wei et al., 2026; Lin et al., 2026b; Wu et al., 2026; Chen et al., 2026a; Tan et al., 2026; Yang et al., 2026; Pan et al., 2026a). In practice, however, good planning and good rendering do not guarantee faithful execution. Models may edit the wrong object, apply the wrong target value, or break the relation that identifies the edited object (Lin et al., 2026a).

We argue that this failure is not explained solely by the capability of either model. It exposes a plan–execution gap at the MLLM–DiT interface. Existing systems commonly compress MLLM outputs

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

into a small set of query tokens, map them with a connector, and use the result as a single cross-attention condition (Lin et al., 2026b; Wei et al., 2026; Pan et al., 2025; Lin et al., 2026a). This treats the edit plan as a single conditioning representation, even though different parts of the plan may need to affect denoising in different ways. This assumption is particularly limiting for instruction-only editing, where no reference image supplies the target appearance as a visual shortcut (Lin et al., 2026b; Jiang et al., 2025; Ku et al., 2024; Wang et al., 2026).

We focus on instruction-only editing, without a reference image or other user-provided spatial control, because this setting removes the visual and spatial shortcuts that can otherwise mask interface failures. Reference-based or spatially guided editors can obtain direct evidence about the target appearance or edit region from an auxiliary condition, while instruction-only editing provides no such shortcut (Lin et al., 2026b; Jiang et al., 2025; Ku et al., 2024; Wang et al., 2026; Pan et al., 2026b). The user supplies only a source video and a natural-language instruction, and the system must infer what the edited state should look like, which object should change, and what value it should take. This makes instruction-only editing both a harder and a more diagnostic setting for studying the plan–execution gap, and a practically attractive interaction paradigm that removes the burden of constructing masks, depth maps, reference images, or other guidance signals from the user.

An edit plan contains at least two complementary kinds of information. *Target-state information* specifies what the edited visual state should become, including source-aware structure and appearance. *Relational constraints* specify which object should change, which attribute should be modified, what value it should take, and which spatial or semantic relation binds these variables. The former should interact directly with video tokens during denoising, while the latter should remain explicit guidance. Collapsing both into one condition obscures this distinction.

We introduce **DUAL**, an MLLM–DiT interface that routes these two types of information through different paths. The Target-State Path aggregates query representations from multiple intermediate and late MLLM layers, projects them with a lightweight module, and concatenates them with source and noisy target tokens. These tokens participate in DiT self-attention and directly shape denoising. The Relational-Constraint Path uses queries from the final MLLM layer together with the native text encoder condition. A small bidirectional refiner produces a residual update to the native text tokens, controlled by a scalar gate initialized at zero, and the resulting condition is used by the DiT's native cross-attention. Thus, DUAL improves how a simple instruction is executed without asking the user to supply additional control signals; the complexity is handled inside the MLLM–DiT interface.

Our design follows three principles: target-state information should interact directly with video tokens during denoising, relational constraints should preserve the native text condition and provide stable guidance, and added conditions should be introduced in a way that preserves the pretrained renderer at initialization. We evaluate the resulting interface on IVEBench and OpenVEBench (Chen et al., 2026b; He et al., 2026), and ablate each path to test whether the two provide complementary benefits. As Figure 1 previews and Section 5 details, DUAL reaches the best Overall score among the compared open methods on both benchmarks, 0.67 on IVEBench and 4.26 on OpenVEBench, while running up to 15.2 times faster than the slowest compared baseline in average inference latency.

Our contributions are:

- We identify collapsing heterogeneous edit-plan information into a single conditioning representation as an important contributor to the plan–execution gap, since target-state information and relational constraints need different mechanisms during denoising.
- We propose DUAL, which routes target-state information into DiT self-attention and relational constraints into cross-attention, via identity position encoding and a zero-initialized residual update that preserve the pretrained renderer.
- DUAL achieves the best Overall score among open methods on both IVEBench and OpenVEBench, and layer-selection and path-removal ablations trace the gains back to the two-path principle.

2 Related Work

Instruction-based video editing. Recent video editors combine pretrained video diffusion backbones with language or multimodal encoders (Lin et al., 2026b; Rao et al., 2026; Bai et al., 2026; Wu

et al., 2025b; Team, 2025; He et al., 2026). Kiwi-Edit (Lin et al., 2026b) uses learnable queries and reference-aware conditioning to improve instruction following and reference fidelity. InsEdit (Rao et al., 2026) studies data-efficient adaptation of video diffusion models for instruction-based editing. These systems motivate simple language-driven interaction, but auxiliary references or spatial controls can provide useful shortcuts when available (Lin et al., 2026b; Jiang et al., 2025; Ku et al., 2024; Wang et al., 2026; Pan et al., 2026b). DUAL instead focuses on the harder instruction-only setting: the user supplies only a source video and a text instruction, and the MLLM–DiT interface must execute the edit without requiring a mask, depth map, reference image, or other manually constructed condition.

MLLM–diffusion interfaces. Unified autoregressive–diffusion systems show that MLLM representations can guide high-fidelity synthesis through feature alignment and token interaction (Shen et al., 2025; Shi et al., 2026b; Pan et al., 2025; Wei et al., 2026; Chen et al., 2026a; Wu et al., 2026). BiVidGen (Ding et al., 2026) reports gains from explicit visual tokens and multi-layer conditioning over opaque latent bridging, and MMCORE (Li et al., 2026) aligns learned query representations with a frozen visual semantic space to counteract the information bottleneck of a fixed query budget. While prior work explores richer alignment, multi-layer fusion, and token-interaction mechanisms, heterogeneous planner information is typically not explicitly routed according to its execution role during denoising. DUAL instead separates target-state information from relational constraints and assigns them distinct interaction paths.

Semantic planning and diffusion editing. Bernini (Team et al., 2026) separates semantic planning from pixel rendering and retains text and source visual features as auxiliary conditioning. Work on asynchronous token decoding for diffusion editing (Shi et al., 2026a) emphasizes that diffusion editing updates tokens through globally coupled dynamics, motivating conditions that can participate directly in token interaction. DUAL builds on both observations: it makes target-state information part of this interaction, while retaining relational constraints as stable guidance.

3 Understanding the Plan–Execution Gap

This section motivates DUAL through three steps: diagnosing a selective, execution-critical degradation (Section 3.2), tracing it to the hierarchical structure of MLLM representations (Section 3.3), and translating these findings into interface design principles (Section 3.4) implemented in Section 4.

3.1 Problem Formulation

Let x_s denote a source video and u an editing instruction, with desired edited video x_t. An MLLM planner produces query states H_Q from (x_s, u), and a DiT renderer predicts the flow-matching velocity of a noisy target latent. A conventional single-bridge system computes

$$c = g(H_Q, u), \qquad \hat{v}_t = F_\theta(z_t, z_s, c, \sigma), \tag{1}$$

where g is a connector, c is consumed as one homogeneous condition (Lin et al., 2026b; Wei et al., 2026; Lin et al., 2026a), z_t and z_s are the noisy target and source latents at noise level σ, F_θ is the DiT renderer, and $\hat{v}_t$ is its predicted velocity. We ask whether this interface preserves the variables needed for faithful execution.

For an instruction such as "turn the cup on the left blue," faithful execution requires the tuple (o, a, v, r): edited object o, attribute a, target value v, and relation or role r (Lin et al., 2026a). An output can have the correct edit type while failing on any of these variables, so we distinguish coarse intent from execution-critical semantics: instruction-compliance and semantic-consistency scores respond to (o, a, v, r), whereas aggregate quality scores can stay high even when these variables are wrong (Chen et al., 2026b; He et al., 2026).

3.2 Diagnosing Selective Degradation

Diagnostic probing of a representative single-bridge system (a learnable-query connector feeding DiT cross-attention) shows that the connector transformation strongly reshapes the representation space, yet probe accuracy for the edited attribute slot stays nearly unchanged while probe scores for

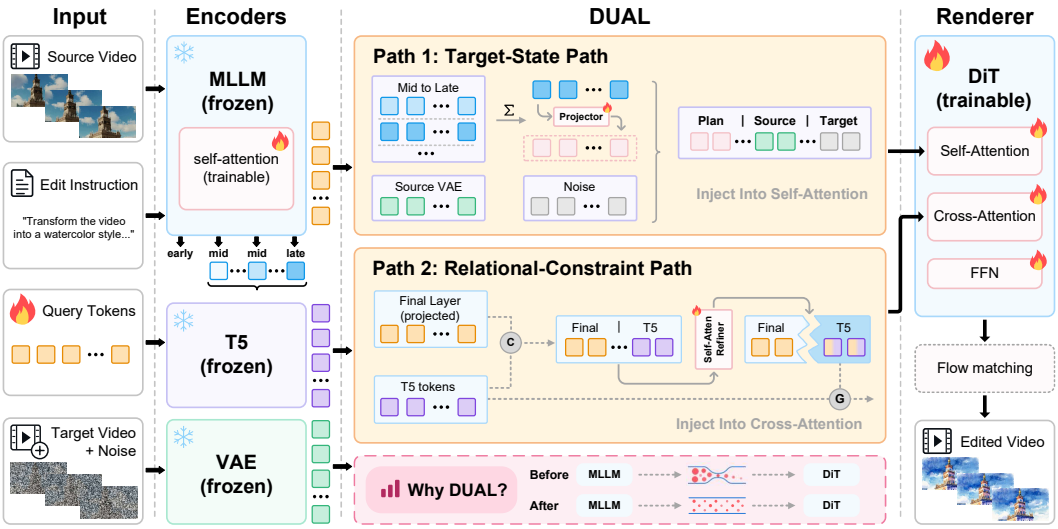


Figure 2: DUAL architecture. The MLLM planner produces query tokens that are routed through the Target-State Path into DiT self-attention and through the Relational-Constraint Path into the native text condition for cross-attention, alongside the frozen VAE and trainable DiT renderer.

target value and edited object drop more sharply (Lin et al., 2026a). This is *selective* degradation, not a uniform loss of information. At the task level, the same setting reports a high structural (wrong-object or wrong-binding) error rate alongside a low under-edit rate: the system is not failing to act, but frequently acting on the wrong variable. This pattern motivates the hypothesis that the connector interface is an important contributor to the plan–execution gap; Section 5 evaluates this hypothesis through controlled ablations. Good planning and good rendering therefore do not guarantee faithful execution.

3.3 From Hierarchical Representations to Different Roles in Denoising

An edit plan contains different types of information. Analyses of autoregressive hidden states in unified multimodal models show that intermediate layers retain more spatial structure, texture, and object-level visual evidence, while later layers encode more abstract, high-level semantics and instruction reasoning, and aggregating both is more effective than using either alone (Shen et al., 2025). This motivates distinguishing two execution roles within a single edit plan:

- **Target-state information:** the visual state toward which the source should evolve, including source-aware structure and local appearance;
- **Relational constraints:** the object, operation, attribute, value, and relation that must remain correct throughout generation.

The source VAE already encodes the *current* state ("what is there now"); target-state information instead specifies the *desired* state ("what should be there after the edit"), so the two are complementary rather than redundant.

Richer multimodal representations are better consumed as part of token interaction than as a cross-attention condition alone: switching from cross-attention to in-context (self-attention) conditioning for a stronger multimodal encoder yields consistent gains (Shi et al., 2026b), consistent with diffusion editing being a token-indexed but globally coupled state-update process in which source-aware visual evidence can participate directly (Shi et al., 2026a). Cross-attention itself, however, is not the problem: systems that anchor a semantic plan to a target representation and feed it through cross-attention alongside text and source visual detail work well in practice when the conditioning is explicit and multi-layer rather than an opaque latent bridge (Ding et al., 2026; Team et al., 2026). The problem is using one representation for information that serves different roles: target-state information should participate in generation, while relational constraints should guide it.

The hypothesis tested by DUAL is therefore not that self-attention is universally preferable to cross-attention, but that different information should enter the DiT in ways that match how it is used during denoising; Section 5 tests this by ablating which MLLM layers feed the Target-State Path.

3.4 Design Principles for the MLLM–DiT Interface

The above observations distill into three principles that DUAL implements.

Preserve the native text condition. Deriving relational constraints solely from a fixed-size set of MLLM queries risks losing lexical detail and object-attribute binding, consistent with the information-bottleneck argument for fixed query budgets (Li et al., 2026; Lin et al., 2026a). DUAL therefore augments, rather than replaces, the native text encoder condition.

Use different processing for the two paths. The Target-State Path uses a lightweight projector, since DiT self-attention performs the fusion with source and target tokens; the Relational-Constraint Path instead combines MLLM queries with native text embeddings through a joint refinement step, in line with unified condition-encoding designs that jointly process text and visual features before fusion (Shen et al., 2025).

Preserve the pretrained renderer at initialization. Relational information from the MLLM should not overwrite the DiT's pretrained text conditioning at the start of training, so DUAL injects it as a residual gated by a scalar initialized at zero.

Closing the plan–execution gap therefore requires more than a stronger connector: the interface should let target-state information participate in generation while relational constraints guide it, rather than reducing the edit plan to one condition.

4 DUAL

Given only a source video and an editing instruction, DUAL routes the resulting edit plan through a Target-State Path and a Relational-Constraint Path, described in turn below, followed by the training objective that supervises both jointly. Figure 2 gives an overview of the architecture.

4.1 Target-State Path

We extract query-placeholder states from MLLM layers $\{20, 24, 28, 32\}$ and aggregate them as

$$H_{\text{state}} = \frac{1}{4} \sum_{\ell \in \{20,24,28,32\}} H_Q^{(\ell)}. \tag{2}$$

The implementation uses the corresponding hidden-state indices after the embedding output. A lightweight projector using RMSNorm (Zhang & Sennrich, 2019) maps these states to the DiT width:

$$P = W_2\, \phi(W_1\, \mathrm{RMSNorm}(H_{\text{state}})) + E_{\text{slot}}, \tag{3}$$

where W_1 and W_2 are learned linear projections, ϕ is a nonlinearity, and E_{slot} is a learned slot embedding added to each plan token. The projector is intentionally lightweight; the following DiT self-attention blocks perform the fusion. The plan tokens are concatenated with source and target tokens and participate in every selected self-attention block:

$$P \leftrightarrow Z_s \leftrightarrow Z_t^{\sigma}. \tag{4}$$

Plan tokens have no physical video coordinates. We therefore apply identity RoPE (Su et al., 2024) to their query and key states and use a learned slot embedding for token identity. Source and target tokens use the native local 3D RoPE (Wan et al., 2025). To distinguish their roles despite sharing the same video grid, we additionally apply source-id rotary phases, with source id 1 and target id 0. This encoding expresses the distinction between "what is there now" and "what is being denoised" without assigning fictitious spatiotemporal locations to the plan.

Table 1: IVEBench results. Params denotes the DiT (renderer) parameter count. IC: Instruction Compliance; SC/BC: Subject/Background Consistency; TF: Temporal Flickering; MS: Motion Smoothness; OSC/PSC: Overall/Phrase Semantic Consistency; IS: Instruction Satisfaction; QA: Quantity Accuracy; Sem./Mot./Cont. Fid.: Semantic/Motion/Content Fidelity. External numbers are quoted from published evaluations under the same protocol. Bold marks the best Overall score; underline marks a tie.

Method	Params	Overall ↑	Vid. Qual. ↑	IC ↑	Vid. Fid. ↑	SC ↑	BC ↑	TF ↑	MS ↑
Ditto (Bai et al., 2026)	14B	<u>0.67</u>	0.78	0.49	0.73	0.96	0.98	0.97	0.99
Bernini (Team et al., 2026)	14B	<u>0.67</u>	0.79	0.48	0.72	0.94	0.96	0.97	0.98
VACE (Jiang et al., 2025)	14B	0.63	0.80	0.25	0.83	0.95	0.97	0.98	0.98
ICVE (Liao et al., 2025)	13B	0.60	0.71	0.45	0.64	0.95	0.97	0.99	1.00
StableV2V (Liu et al., 2026)	1.4B	0.51	0.69	0.43	0.41	0.85	0.92	0.96	0.96
Omni-Video (Tan et al., 2026)	1.3B	0.59	0.78	0.44	0.54	0.96	0.97	0.98	0.99
AnyV2V (Ku et al., 2024)	1.3B	0.58	0.73	0.42	0.59	0.89	0.94	0.97	0.97
InsV2V (Cheng et al., 2023)	0.9B	<u>0.67</u>	0.80	0.39	0.82	0.94	0.96	0.97	0.97
LucyEdit-Dev (Team, 2025)	5B	0.64	0.82	0.34	0.75	0.95	0.96	0.98	0.99
DUAL (ours)	5B	**0.67**	0.84	0.37	0.79	0.95	0.97	0.98	0.99

Method	VTSS ↑	OSC ↑	PSC ↑	IS ↑	QA ↑	Sem. Fid. ↑	Mot. Fid. ↑	Cont. Fid. ↑
Ditto (Bai et al., 2026)	0.04	0.25	0.24	3.87	0.30	0.89	0.79	3.64
Bernini (Team et al., 2026)	0.04	0.24	0.23	4.23	0.00	0.91	0.45	3.99
VACE (Jiang et al., 2025)	0.04	0.24	0.22	2.16	0.20	0.97	0.89	4.03
ICVE (Liao et al., 2025)	0.02	0.23	0.23	3.62	0.30	0.85	0.46	3.55
StableV2V (Liu et al., 2026)	0.02	0.20	0.24	3.56	0.20	0.70	0.75	1.79
Omni-Video (Tan et al., 2026)	0.04	0.22	0.23	3.36	0.40	0.81	0.51	2.85
AnyV2V (Ku et al., 2024)	0.03	0.22	0.24	3.34	0.30	0.80	0.82	2.75
InsV2V (Cheng et al., 2023)	0.04	0.24	0.23	3.06	0.30	0.95	0.86	4.05
LucyEdit-Dev (Team, 2025)	0.05	0.24	0.22	2.84	0.20	0.93	0.68	3.83
DUAL (ours)	0.06	0.24	0.22	3.09	0.20	0.97	0.50	4.33

4.2 Relational-Constraint Path

Queries from the final MLLM layer, $H_Q^{(L)}$, are projected to the native text width and combined with frozen native text instruction tokens T:

$$Q = W_q H_Q^{(L)}, \qquad R = \mathrm{BiTransformer}([Q; T]), \tag{5}$$

where W_q is a learned linear projection and R is the refiner's output sequence over the concatenated queries and text tokens. The refiner has two bidirectional Transformer layers. Its sub-sequence of R corresponding to the text positions produces a residual update Δ_T. We preserve the pretrained text condition with

$$C_{\mathrm{sem}} = T + \alpha \Delta_T, \qquad \alpha(0) = 0, \tag{6}$$

where C_{sem} is the refined text condition and α is a scalar gate initialized at zero. The refined condition is passed through the native DiT text-embedding projection and cross-attention. Thus the native text encoder preserves instruction detail, while MLLM queries add object and relation information from the source video.

4.3 Training Objective

For flow matching (Lipman et al., 2023), we sample a noise level σ, construct the noisy target latent z_t^σ, and supervise only the target output with the native velocity objective:

$$\mathcal{L}_{\mathrm{flow}} = \mathbb{E}\left[\|\hat{v}_t - v_t\|_2^2\right], \tag{7}$$

where $\hat{v}_t$ is the predicted velocity and v_t is the ground-truth flow-matching velocity for the target sample. The two paths are trained jointly through the target-only flow objective. We additionally

Table 2: OpenVEBench results. Params denotes the DiT (renderer) parameter count. † denotes a closed-source system, listed for reference only and excluded from Overall-score ranking. Ranking formatting is applied only to the displayed Overall scores among open methods.

Method	Params	Overall ↑	Glob. Style ↑	Back-gr. ↑	Loc. Change ↑	Loc. Remove ↑	Loc. Add ↑	Sub-title ↑	Cre-ative ↑	Cam-era ↑
Runway Aleph†	–	4.49	4.41	4.40	4.31	4.64	4.36	4.21	4.81	4.70
Bernini (Team et al., 2026)	14B	<u>4.17</u>	4.64	3.77	4.06	4.68	3.51	3.99	4.18	4.84
Ditto (Bai et al., 2026)	14B	3.44	4.48	3.52	2.89	3.53	2.48	3.69	4.14	3.33
VACE (Jiang et al., 2025)	14B	3.01	3.46	2.81	2.47	3.99	1.76	4.41	2.17	3.09
ICVE (Liao et al., 2025)	13B	3.76	3.87	3.51	3.87	4.50	3.77	4.68	3.54	2.84
InsViE (Wu et al., 2025b)	2B	3.25	3.63	2.68	2.82	3.56	2.25	4.77	3.36	3.61
Omni-Video (Tan et al., 2026)	1.3B	3.66	3.41	4.11	3.75	4.52	2.80	4.95	1.13	3.62
OpenVE-Edit (He et al., 2026)	5B	3.89	4.24	4.10	3.80	3.50	3.41	3.98	3.71	3.25
LucyEdit (Team, 2025)	5B	3.77	3.64	3.25	3.93	3.95	3.92	4.23	4.19	3.54
Kiwi-Edit (Lin et al., 2026b)	5B	3.88	4.47	3.60	3.85	4.32	3.31	4.11	3.88	3.57
DUAL (ours)	5B	**4.26**	3.90	3.94	4.44	4.50	4.43	4.96	3.68	3.84

apply high-noise loss reweighting: the standard bell-shaped timestep weight is boosted by an additive $2.0\sigma^4$ term at high noise levels. Because high-noise timesteps correspond to coarse denoising stages where object identity and layout are largely determined, this reweighting increases the training signal at the stages most relevant to object and binding errors.

Training curriculum. We train DUAL with a three-stage curriculum using mixed open-source text-to-video (t2v), video-to-video (v2v), and image-to-image (i2i) data, together with internal v2v data. Image editing first establishes precise edit grounding, mixed image/video training then transfers this grounding into temporal execution, and full-distribution refinement balances editing accuracy, temporal consistency, and generation quality. The complete stage-wise data schedule and its motivation are given in Appendix A.1.

5 Experiments

5.1 Experimental Setup

Benchmarks and metrics. We evaluate instruction-only video editing on IVEBench (Chen et al., 2026b) and OpenVEBench (He et al., 2026). Both benchmarks use a common protocol in which methods receive a source video and an editing instruction, without user-provided masks, depth maps, reference images, or other auxiliary controls. IVEBench reports an aggregate score together with video-quality metrics (Video Quality, Instruction Compliance, Video Fidelity, Subject and Background Consistency, Temporal Flickering, Motion Smoothness, VTSS) and semantic metrics (Overall and Phrase Semantic Consistency, Instruction Satisfaction, Quantity Accuracy, and Semantic, Motion, and Content Fidelity). OpenVEBench reports an overall score over eight editing categories. Instruction-satisfaction and fidelity metrics are scored by a fixed VLM judge, and all evaluator prompts and aggregation rules are held constant across methods.

Baselines. On IVEBench, we compare against Ditto (Bai et al., 2026), InsV2V (Cheng et al., 2023), Bernini (Team et al., 2026), Lucy Edit (Team, 2025), VACE (Jiang et al., 2025), ICVE (Liao et al., 2025), Omni-Video (Tan et al., 2026), AnyV2V (Ku et al., 2024), and StableV2V (Liu et al., 2026). On OpenVEBench, we compare against Runway Aleph (Runway, 2025), Bernini (Team et al., 2026), OpenVE-Edit (He et al., 2026), Lucy Edit (Team, 2025), ICVE (Liao et al., 2025), Omni-Video (Tan et al., 2026), Ditto (Bai et al., 2026), InsViE (Wu et al., 2025b), VACE (Jiang et al., 2025), and Kiwi-Edit (Lin et al., 2026b). These baselines cover instruction-guided, reference-guided, and unified video editors, and several condition the renderer through a single connector into cross-attention. The comparison therefore spans the conventional interface and DUAL's two paths across a range of backbone scales.



Figure 3: Side-by-side comparison on object removal, attribute change, style transfer, and color change. Rows: source video, OmniWeaving, Kiwi-Edit, Ditto, Lance, and DUAL. DUAL edits only the target region, preserving lighting and character identity elsewhere.

Implementation details. The Reproducibility Statement and Table 6 give the complete model and inference configuration, while Appendix Table 7 and Appendix A.1 summarize and detail the training data and schedule. All methods are evaluated with their respective reported configurations.

5.2 Main Results

IVEBench and OpenVEBench. Table 1 reports IVEBench results, and Table 2 reports OpenVEBench results across eight editing categories. DUAL reaches the highest Overall score among the compared open methods on both benchmarks, tied at 0.67 on IVEBench and 4.26 on OpenVEBench, exceeding the closest 5B reference, Kiwi-Edit, with the margin concentrated in local edits such as Local Add. Kiwi-Edit represents the conventional interface, which compresses planner output into a single cross-attention condition, so this margin is consistent with the intended role of the two paths: target-state information interacts with video tokens in self-attention, while relational constraints remain in the text condition for cross-attention guidance. The fine-grained tables show that DUAL achieves this without a large increase in parameter count, though individual baselines remain stronger on specific consistency or compliance metrics.

Qualitative comparison. Figure 3 places OmniWeaving (Pan et al., 2026a), Kiwi-Edit (Lin et al., 2026b), Ditto (Bai et al., 2026), Lance (Fu et al., 2026), and DUAL side by side on cases covering object removal, attribute change, style transfer, and color change, deliberately selected because other methods perform an edit but bind it to the wrong object, region, or style, the failures the two paths are meant to address. Across these cases, DUAL applies the requested change only to the target region while keeping lighting and character identity consistent with the source, whereas other methods alter the wrong region or fail to preserve the intended edit.

5.3 Ablations

Table 3 isolates each design decision from Section 3.4 on OpenVEBench, and Table 4 summarizes which components each variant keeps; all variants share the configuration in Table 7.

Necessity of both paths. Table 3 shows that removing either path, or both (w/o DUAL, which conditions only on the native text embeddings), lowers Overall relative to Full DUAL, with the largest category-level drops falling on different categories depending on which path is removed. This non-uniform pattern is consistent with the two paths contributing complementary rather than redundant information.

Table 3: Ablations on OpenVEBench. Each group isolates one design decision; the last row of each group is the DUAL (default) setting.

Variant	Overall ↑	Glob. Style ↑	Back-gr. ↑	Loc. Change ↑	Loc. Remove ↑	Loc. Add ↑	Sub-title ↑	Cre-ative ↑	Cam-era ↑
Execution paths									
w/o DUAL	3.23	2.27	2.63	3.52	3.21	3.40	4.04	3.31	3.59
w/o Target-State Path	3.44	2.26	3.64	3.92	3.10	3.63	3.90	3.29	3.75
w/o Relational-Constraint Path	3.40	2.42	3.67	3.88	3.19	3.27	3.88	3.21	3.72
Full DUAL (default)	**4.26**	**3.90**	**3.94**	**4.44**	**4.50**	**4.43**	**4.96**	**3.68**	**3.84**
Layer selection									
Early-to-mid $\{4, 8, 12, 16\}$	3.32	2.28	3.65	3.71	3.02	3.15	3.88	3.21	3.72
Final layer only	3.45	2.28	3.60	3.95	3.22	3.52	3.96	3.28	3.73
Mid-to-late $\{20, 24, 28, 32\}$ (default)	**4.26**	**3.90**	**3.94**	**4.44**	**4.50**	**4.43**	**4.96**	**3.68**	**3.84**

Table 4: Ablation coverage. Each row is a variant from Table 3; each column is a design component. TSP/RCP: Target-State/Relational-Constraint Path. ✓ marks a component present in that variant, ✗ marks a component removed or replaced.

Variant	TSP	RCP	Layers
w/o DUAL	✗	✗	mid-late
w/o TSP	✗	✓	mid-late
w/o RCP	✓	✗	mid-late
Full DUAL (default)	✓	✓	mid-late
Early-to-mid	✓	✓	early-mid
Final layer only	✓	✓	final
Mid-to-late (default)	✓	✓	mid-late

Table 5: Efficiency comparison on IVE. Params denotes the DiT (renderer) parameter count. Latency denotes average inference time in seconds.

Method	Params	Latency (s)
Bernini	14B	893
OmniWeaving	8.3B	376
Ditto	14B	2385
Kiwi-Edit	5B	168
DUAL (ours)	5B	157

Layer selection. We compare using only the final MLLM layer, an early-to-mid layer set $\{4, 8, 12, 16\}$, and our default mid-to-late aggregation $\{20, 24, 28, 32\}$ for the Target-State Path. The default reaches the highest Overall, supporting mid-to-late layers as the better source for the target-state representation in this system.

5.4 Efficiency Analysis

Inference latency. The Target-State Path appends 512 plan tokens to the DiT sequence, so DUAL attends over a longer sequence than an interface that conditions the renderer through cross-attention alone. Table 5 reports the average IVE inference latency and parameter count for each method. DUAL takes 157 seconds at 5B parameters, compared with 168 seconds for Kiwi-Edit at the same parameter scale, while remaining faster than Bernini (893 seconds), OmniWeaving (376 seconds), and Ditto (2385 seconds) in this comparison.

6 Conclusion

We presented DUAL, an MLLM–DiT interface for instruction-only video editing that routes target-state information into DiT self-attention and relational constraints into the native cross-attention condition, rather than compressing both into one condition sequence. The gains concentrate in instruction compliance and local editing, and removing either path harms different metrics, consistent with their complementary roles. These results support the connector interface as a contributing factor, not a proven sole cause, of the plan–execution gap, and do not claim self-attention is categorically superior to cross-attention.

AI USE STATEMENT

We used generative AI tools solely to improve the writing, grammar, phrasing, and readability of this manuscript. All technical content, research ideas, experimental design, analysis, and conclusions were developed and verified by the authors.

ETHICS STATEMENT

This work uses existing video-editing data and does not involve human subjects or collection of personally identifiable information.

REPRODUCIBILITY STATEMENT

DUAL uses publicly available pretrained components: Wan2.2 TI2V-5B as the renderer, Qwen2.5-VL-3B-Instruct as the planner, and a frozen UMT5-XXL text encoder. The method section specifies the interface design, including layer selection, token routing, positional handling, and gating. Table 6 (Appendix A.1) consolidates the model, training, inference, and evaluation settings needed to reproduce the reported results. The detailed training-data schedule is given separately in Appendix Table 8. Each ablation in Table 3 changes only the named component from the default configuration. Reported latency is the average per-sample wall-clock time for instruction-only inference on the same 8-GPU setup.

REFERENCES

Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, Yinghao Xu, Yujun Shen, and Qifeng Chen. Scaling instruction-based video editing with a high-quality synthetic dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 37971–37981, June 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Junyi Chen, Tong He, Zhoujie Fu, Pengfei Wan, Kun Gai, and Weicai Ye. Vino: A unified visual generator with interleaved omnimodal context, 2026a. URL https://arxiv.org/abs/2601.02358.

Yinan Chen, Jiangning Zhang, Teng Hu, Yuxiang Zeng, Zhucun Xue, Qingdong He, Chengjie Wang, Yong Liu, Xiaobin Hu, and Shuicheng Yan. Ivebench: Modern benchmark suite for instruction-guided video editing assessment. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Jiaxin Cheng, Tianjun Xiao, and Tong He. Consistent video-to-video transfer using synthetic dataset, 2023. URL https://arxiv.org/abs/2311.00213.

Yanbo Ding, Yijia Fan, Caihua Shan, Yifan Yang, Yifei Shen, Weijie Wang, Xirui Hu, Dongsheng Li, Lili Qiu, Yuqing Yang, and Yali Wang. Beyond text conditioning: A systematic study of mllm-dit fusion for video generation, 2026. URL https://arxiv.org/abs/2608.14043.

Fengyi Fu, Mengqi Huang, Shaojin Wu, Yunsheng Jiang, Yufei Huo, Hao Li, Yinghang Song, Fei Ding, Jianzhu Guo, Qian He, Zheren Fu, Zhendong Mao, and Yongdong Zhang. Lance: Unified multimodal modeling by multi-task synergy, 2026. URL https://arxiv.org/abs/2605.18678.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, Itay Chachy, Jonathan Chetboun, Michael Finkelson, Michael Kupchick, Nir Zabari, Nitzan Guetta, Noa Kotler, Ofir Bibi, Ori Gordon, Poriya Panet, Roi Benita, Shahar Armon, Victor Kulikov, Yaron Inger, Yonatan Shiftan, Zeev Melumian, and Zeev Farbman. Ltx-2: Efficient joint audio-visual foundation model, 2026. URL https://arxiv.org/abs/2601.03233.

Haoyang He, Jie Wang, Jiangning Zhang, Zhucun Xue, Xingyuan Bu, Qiangpeng Yang, Shilei Wen, and Lei Xie. Openve-3m: A large-scale high-quality dataset for instruction-guided video editing, 2026. URL https://arxiv.org/abs/2512.07826.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17191–17202, October 2025.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025. URL https://arxiv.org/abs/2412.03603.

Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. Anyv2v: A tuning-free framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024.

Zijie Li, Yichun Shi, Jingxiang Sun, Ye Wang, Yixuan Huang, Zhiyao Guo, Xiaochen Lian, Peihao Zhu, Yu Tian, Zhonghua Zhai, and Peng Wang. Mmcore: Multimodal connection with representation aligned latent embeddings, 2026. URL https://arxiv.org/abs/2604.19902.

Xinyao Liao, Xianfang Zeng, Ziye Song, Zhoujie Fu, Gang Yu, and Guosheng Lin. In-context learning with unpaired clips for instruction-based video editing, 2025. URL https://arxiv.org/abs/2510.14648.

Hangyu Lin, Chao Wen, Chengming Xu, Jianxiong Gao, Jiangning Zhang, Xiaobin Hu, and Yanwei Fu. What semantics survive the connector? diagnosing vlm-to-dit alignment in video editing, 2026a. URL https://arxiv.org/abs/2605.20795.

Yiqi Lin, Guoqiang Liang, Ziyun Zeng, Zechen Bai, Yanzhe Chen, and Mike Zheng Shou. Kiwi-edit: Versatile video editing via instruction and reference guidance, 2026b. URL https://arxiv.org/abs/2603.02175.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Chang Liu, Rui Li, Kaidong Zhang, Yunwei Lan, and Dong Liu. Stablev2v: Stabilizing shape consistency in video-to-video editing. *IEEE Transactions on Circuits and Systems for Video Technology*, 36(4):5467–5482, April 2026. ISSN 1558-2205. doi: 10.1109/tcsvt.2025.3639307. URL http://dx.doi.org/10.1109/TCSVT.2025.3639307.

Kaihang Pan, Qi Tian, Jianwei Zhang, Weijie Kong, Jiangfeng Xiong, Yanxin Long, Shixue Zhang, Haiyi Qiu, Tan Wang, Zheqi Lv, Yue Wu, Liefeng Bo, Siliang Tang, and Zhao Zhong. Omniweaving: Towards unified video generation with free-form composition and reasoning, 2026a. URL https://arxiv.org/abs/2603.24458.

Tianlin Pan, Jiayi Dai, Chenpu Yuan, Zhengyao Lv, Binxin Yang, Hubery Yin, Chen Li, Jing Lyu, Caifeng Shan, and Chenyang Si. Nova: Sparse control, dense synthesis for pair-free video editing, 2026b. URL https://arxiv.org/abs/2603.02802.

Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, Ji Hou, and Saining Xie. Transfer between modalities with metaqueries, 2025. URL https://arxiv.org/abs/2504.06256.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Zhefan Rao, Bin Zou, Haoxuan Che, Xuanhua He, Chong Hou Choi, Yanheng Li, Rui Liu, and Qifeng Chen. Insedit: Towards instruction-based visual editing via data-efficient video diffusion models adaptation, 2026. URL https://arxiv.org/abs/2604.08646.

Runway. Runway aleph: A new way to edit, transform and generate video., 2025. URL https://runway.com/research/introducing-runway-aleph.

Tao Shen, Xin Wan, Taicai Chen, Rui Zhang, Junwen Pan, Dawei Lu, Fanding Lei, Zhilin Lu, Yunfei Yang, Chen Cheng, Qi She, Chang Liu, and Zhenbang Sun. Mammothmoda2: A unified ar-diffusion framework for multimodal understanding and generation, 2025. URL https://arxiv.org/abs/2511.18262.

Yang Shi, Liangsi Lu, Minzhe Guo, Yifeng Xie, Yanhui Chen, Jingchao Wang, and Xuhang Chen. Diffusion image editing via asynchronous token decoding, 2026a. URL https://arxiv.org/abs/2608.09322.

Yangming Shi, Shixiang Zhu, Tao Shen, Zhimiao Yu, Dengsheng Chen, Taicai Chen, Yunfei Yang, Juan Zhou, Chen Cheng, Liang Ma, Xibin Wu, Benxuan Yan, Ge Li, Tuoyu Zhang, Dan Li, Chang Liu, and Zhenbang Sun. Mamoda2.5: Enhancing unified multimodal model with dit-moe, 2026b. URL https://arxiv.org/abs/2605.02641.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Zhiyu Tan, Hao Yang, Luozheng Qin, Jia Gong, Mengping Yang, and Hao Li. Omni-video: Democratizing unified video understanding and generation, 2026. URL https://arxiv.org/abs/2507.06119.

Bernini Team, Chenchen Liu, Junyi Chen, Lei Li, Lu Chi, Mingzhen Sun, Zhuoying Li, Yi Fu, Ruoyu Guo, Yiheng Wu, Ge Bai, and Zehuan Yuan. Bernini: Latent semantic planning for video diffusion, 2026. URL https://arxiv.org/abs/2605.22344.

DecartAI Team. Lucy edit: Open-weight text-guided video editing, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Tong Wang, Meng Zou, Chengjing Wu, Xiaochao Qu, Luoqi Liu, Xiaolin Hu, and Ting Liu. Mive: Multiscale vision-language features for reference-guided video editing, 2026. URL https://arxiv.org/abs/2605.14664.

Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhu Chen. Univideo: Unified understanding, generation, and editing for videos, 2026. URL https://arxiv.org/abs/2510.08377.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, Linus, Patrol, Peizhen Zhang, Peng Chen, Penghao Zhao, Qi Tian, Songtao Liu, Weijie Kong, Weiyan Wang, Xiao He, Xin Li, Xinchi Deng, Xuefei Zhe, Yang Li, Yanxin Long, Yuanbo Peng, Yue Wu, Yuhong Liu, Zhenyu Wang, Zuozhuo Dai, Bo Peng, Coopers Li, Gu Gong, Guojian Xiao, Jiahe Tian, Jiaxin Lin, Jie Liu, Jihong Zhang, Jiesong Lian, Kaihang Pan, Lei Wang, Lin Niu, Mingtao Chen, Mingyang Chen, Mingzhe Zheng, Miles Yang, Qiangqiang Hu, Qi Yang, Qiuyong Xiao, Runzhou Wu, Ryan Xu, Rui Yuan, Shanshan Sang, Shisheng Huang, Siruis Gong, Shuo Huang, Weiting Guo, Xiang Yuan, Xiaojia Chen, Xiawei Hu, Wenzhi Sun, Xiele Wu, Xianshun Ren, Xiaoyan Yuan, Xiaoyue Mi, Yepeng Zhang, Yifu Sun, Yiting Lu, Yitong Li, You Huang, Yu Tang, Yixuan Li, Yuhang Deng, Yuan Zhou, Zhichao Hu, Zhiguang Liu, Zhihe Yang, Zilin Yang, Zhenzhi Lu, Zixiang Zhou, and Zhao Zhong. Hunyuanvideo 1.5 technical report, 2025a. URL https://arxiv.org/abs/2511.18870.

Jianzong Wu, Hao Lian, Jiongfan Yang, Dachao Hao, Ye Tian, Yunhai Tong, Jingyuan Zhu, Biaolong Chen, Qiaosong Qi, Aixi Zhang, Wanggui He, Mushui Liu, Jinlong Liu, Pipei Huang, and Hao Jiang. Loomvideo: Unifying multimodal inputs into video generation and editing, 2026. URL https://arxiv.org/abs/2606.06042.

Yuhui Wu, Liyi Chen, Ruibin Li, Shihao Wang, Chenxi Xie, and Lei Zhang. Insvie-1m: Effective instruction-based video editing with elaborate dataset construction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16692–16701, October 2025b.

Hao Yang, Zhiyu Tan, Jia Gong, Luozheng Qin, Hesen Chen, Xiaomeng Yang, Yuqing Sun, Yuetan Lin, Mengping Yang, and Hao Li. Omni-video 2: Scaling mllm-conditioned diffusion for unified video generation and editing, 2026. URL https://arxiv.org/abs/2602.08820.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1e8a19426224ca89e83cef47f1e7f53b-Paper.pdf.

A Appendix

A.1 Implementation Details

This appendix records the training schedule; the remaining model and run settings are consolidated in Table 6.

We use a three-stage curriculum that progressively moves from edit grounding to temporal execution and joint refinement.

Stage 1: grounding edit semantics. The first stage uses image-to-image (i2i) editing only. By restricting the input to static editing examples, the model first learns the most basic execution-critical correspondence: which object should change, which attribute should be modified, and what the desired target state should look like. This stage establishes a precise semantic starting point before temporal variation is introduced; no text-to-video (t2v) data is used at this stage.

Table 6: Reproducibility settings.

Aspect	Setting
Model selection	Renderer Wan2.2 TI2V-5B; planner Qwen2.5-VL-3B-Instruct; frozen UMT5-XXL text encoder.
Trainable parameters	512 MLLM video queries, all DiT weights, and rank-64 LoRA on MLLM attention projections (q/k/v/o); text encoder, VAE, and remaining MLLM weights frozen.
Optimizer	AdamW, learning rate 1×10^{-5}, constant schedule, 400-step warm-up.
Precision / sharding	bf16 mixed precision with ZeRO stage 2.
Checkpoint averaging	EMA, decay 0.9999; all reported numbers use EMA weights.
Sampling	50 flow-matching steps, seed 0, 15 fps, no classifier-free guidance, one forward pass per step.
Evaluation input	Instruction-only: source video plus text instruction, no mask/depth/reference image.
Resolution / frames	Native resolution, up to 921,600 pixels/frame, full video, same for all DUAL runs.
Ablation protocol	One factor at a time from the default DUAL configuration.
Hardware	Eight GPUs for both training and inference.

Stage 2: lifting editing into time. The second stage combines video-to-video (v2v), i2i, and t2v data. Within v2v, reconstruction pairs are introduced before editing pairs dominate the schedule. Reconstruction first teaches the renderer to preserve content and propagate visual states consistently across frames; the schedule then shifts toward editing pairs, which attach the previously grounded edit semantics to this stable temporal execution mechanism. The i2i fraction decreases as v2v editing becomes dominant, retaining precise spatial and appearance supervision during the transition. The t2v fraction also decreases but remains nonzero as an anchor for the pretrained text-to-video generation prior.

Stage 3: joint refinement. The third stage retains v2v editing, i2i, and t2v, but further concentrates the training signal on v2v editing. At this point, temporal propagation has already been established, so the goal is not to introduce a new task but to jointly refine localized editing, cross-frame consistency, and natural generation under the full task distribution. i2i and t2v remain as small stabilizing streams rather than primary objectives. The training schedule used in our experiments is summarized in Table 8.

Table 7: Training schedule summary. The full run configuration is in Table 6; data proportions by stage are in Table 8.

Component	Setting
Training data	Mixed open-source t2v, v2v, and i2i data, plus internal v2v data
Training stages	Stage 1: 5K steps; Stage 2: 100K steps; Stage 3: 50K steps
Batch size	8×4 (effective batch size 32)

Table 8: Three-stage training curriculum used in our experiments. Percentages denote sampling proportions at the beginning and end of each stage, linearly annealed within each stage; the v2v entry is split into reconstruction (recon.) and editing pairs.

Stage	Purpose	v2v (recon. / edit)	i2i	t2v
1	Ground edit semantics	–	100%	0%
2	Transfer grounding to temporal execution	50% (35 / 15) $\rightarrow$ 70% (7 / 63)	30%$\rightarrow$20%	20%$\rightarrow$10%
3	Joint refinement	70%$\rightarrow$90% (editing only)	20%$\rightarrow$5%	10%$\rightarrow$5%

A.2 LIMITATIONS

Full self-attention over plan tokens increases sequence length and therefore cost, which motivates using fewer plan tokens or sparse attention. DUAL also depends on a planner capable of forming usable target-state information.